

# m-learning, mobile experimentation and telepresence with cell phones and PDAs

A.Bischoff<sup>1</sup>

<sup>1</sup> University of Hagen, Control engineering group, Hagen, Germany

**Abstract**—Mobile devices such as notebooks and PDAs are very interesting tools for web-based teaching and distant teaching today. We have adapted Web-based remote laboratory environments to mobile devices like PDAs and smartphones to remotely control a Pioneer 3 AT mobile robot.

**Index Terms**—m-learning, telerobotics, distant learning, speech recognition

## I. INTRODUCTION

Since the typical students of distant teaching universities are professionals, mobile learning will give them the opportunity to effectively use also small amounts of free time. In particular the rising offer of campus WLANs context based studying becomes 'on demand learning'. As an example a mobile user is able to do a search in an online encyclopedia to discover information about a building. Beyond the university based teaching a need for lifelong learning exists. Today acquired skills must constantly be extended and updated. In this context mobile learning can point out a rational way. To provide our student with a mobile learning solution two web-based remote laboratory environments have been adapted to mobile devices like PDAs and smartphones.

By the development of wireless communication networks like portable mobile phone networks (GSM, GPRS, EDGE, UMTS) and local wireless networks (WLAN), electronic teaching material can be accessed from any location. Since the typical students of distant teaching universities are professionals, mobile learning will give them the opportunity to effectively use also small amounts of spare time.

## II. ADAPTION OF USER INTERFACES TO MOBILE DEVICES

Since some mobile devices like PDAs and smartphones have screen sizes up to 640x480 pixels, which was the standard VGA resolution of PCs in the 90s, most PDAs today only provides 320x240 pixels. The situation on mobile phones is even worse, resolutions of typical 176x220 pixels have to be taken into account.

Nearly all available mobile web browsers like Pocket IE, Opera, Netfront and Minimo are offering mobile render modes which adapts standard web sites to mobile viewing (scaling of images and text, avoiding of horizontal

scrolling, full screen mode). Nevertheless hand optimizing of mobile interfaces of remote labs is required. In depth knowledge of the target devices (OS and target browser) and heavy testing is important to develop a successful application. In case of very small display sizes vertical scrolling or a tabbed interface have to be accepted. Since some smartphones do not provide any touchscreen functionality, user interfaces usability must be tested also with 'jog dial' input devices. Some of those restrictions will overcome in near future when innovative user interfaces like laser projection keyboards and LED-based projectors will be integrated in mobile devices.

## III. WEB BASED REMOTE LABORATORIES

New and since 2001 existing Web based remote laboratory environments [1] have been adapted to mobile



Figure 1: Omnidirectional mobile robot platform, GUI, experiment

devices like PDAs and smartphones in the remote lab of the University of Hagen.

- An omnidirectional mobile robot platform: Students task is to adapt parameters off the underlying motor control. The complete AWT based experimentation GUI for PC environment can be reused on PDAs and smartphones. The mobile robot moves on preplanned paths (fig. 1)
- A web controlled Pioneer 3 AT mobile robot: A robot remote control application which demonstrate the features of the robot (fig.2)

Since not all implementations of PocketPCs are supporting the Java integration to a web page without of installation of commercial software an alternative solution is required for student usage with their own devices. An environment that works 'out of the box' with common Windows Mobile devices is needed.

#### IV. MOBILE ROBOT

As an example of online mobile remote experimentation a robot remote control application which demonstrate the features of a PIONEER 3 -AT mobile robot was chosen.



Figure 2: Web and Java AWT based GUI on a smartphone (640x480 pixel)

The PIONEER 3-AT is a wheeled all-terrain experimental mobile robot. It can be equipped with various types of

sensors, including front and rear sonar, a laser rangefinder, bumpers, a pan-tilt-zoom color camera and GPS receiver. Robot motion is controlled by an embedded microprocessor, which communicates with a PC client. This communication is used for sending motion commands to the microcontroller and for data transfer between micro controller and a PC client. The PC client is mounted on the robot and running under Linux OS. It is equipped with a wireless radio modem and thereby accessible through our network, which makes it an ideal platform for remote control. The on board PC is able to communicate via wireless LAN (WLAN) with connected computers over the Internet. Delivered software enables to program the robot on various levels, ranging from simple micro controller commands to complex actions.

Benefits of the system are different sensor systems like ultrasonic, laser scanner and a CCD camera system together with an application programmers interface (Aria and Saphira-API) for autonomous behavior. The onboard PC is able to communicate via wireless LAN (WLAN) with connected computers over the Internet. Since a proprietary remote control software is provided by ActivMedia, users with a special client are able to control the robot from distance. To avoid the download and installation of commercial software for the remote clients, a web-based solution is highly demanded. We have developed a client/server (Java application/Java applet) based solution for remote control of the robot and video streaming to the web-based client. The remote control server application is a Java application which uses the Aria-API (a linux library binary and C++ source code) with automatically generated Java wrapper classes (from C++ source by SWIG [2]) for low level control of the drives. A Linux video streaming application and a web server are running directly on the robot's on board PC.

On the client side, only a Java-enabled web browser is required. A live video stream for viewing the robot's camera view helps to remotely control the behavior.

#### V. VIDEO STREAMING

The robots software was supplemented by a Linux based streaming server (ffmpeg/ffserver [10]) with MPEG4 support and a self developed Java server application to realize a web-based remote control with real time video stream feedback. In a first step a PC client based on Java Media Framework and a Java control applet was developed at the university.

To provide the same functionality to a mobile device more effort is required, because no native Java Media Framework support for PDA platforms is available. The bundled Windows Media Player in common Windows Mobile and PocketPC variants (PPC2002, WM2003, WM5, WM6) is capable to playback MPEG4 compressed video streams in full screen mode. Windows CE NET 4.1 and Windows Mobile 2003 up to WM 2006 are supporting embedded Windows Media Player in a web page.

## VI. ASYNCHRONOUS JAVASCRIPT CLIENT

Some Windows CE NET 4.1 based devices, like the bSquare Maui are supporting a Java variant (Personaljava) 'out of the box' (included in ROM), but are not available on the market anymore. These devices are suitable to run an adapted PC control applet [3]. To overcome these restrictions a new approach based on asynchronous Javascript was realized. This asynchronous Javascript client sends user requests (button events) to the server without a reload of the web page. Each reload will stop the video stream initially and has to be avoided.



Figure 3: Pioneer 3AT mobile robot controlled by Windows Mobile 2003 smartphone

On server side this asynchronous behavior was realized with Apache NPH-CGI scripts. NPH means non parsed headers, which allows CGI-scripts to get full control over

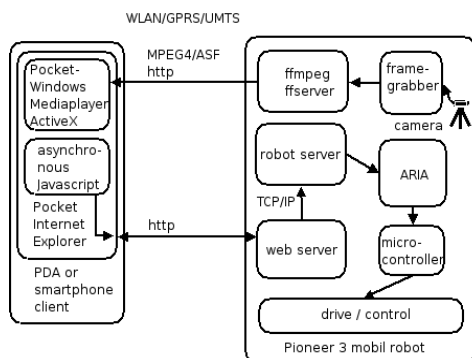


Figure 5: Asynchronous client communication structure

the HTTP protocol response. The HTTP response 204 'no content' (see RFC 2616 [4]) avoid any reload of the current HTML page on client side. A limited or restricted internet connection over a firewall or a proxy with just open port 80 for HTTP access is suitable for the whole functionality. Another additional benefit of this solution compared to a Java client is the HTTP connectionless protocol, which is much more robust to connection interruptions, especially in case of GPRS and UMTS transmissions. Concurrent connections, like a collaborative remote control of the robot are also possible

with a stateless protocol. In this case the CGI-scripts sends command locally to the Java server component, a transfer over an extra TCP port is not necessary.

With asynchronous Javascript an application like behavior like a Java or a Flash can be established. In combination with the exchange of XML based messages between client and server these techniques is well known under the term 'AJAX' [5]. Windows Mobile does not support an AJAX

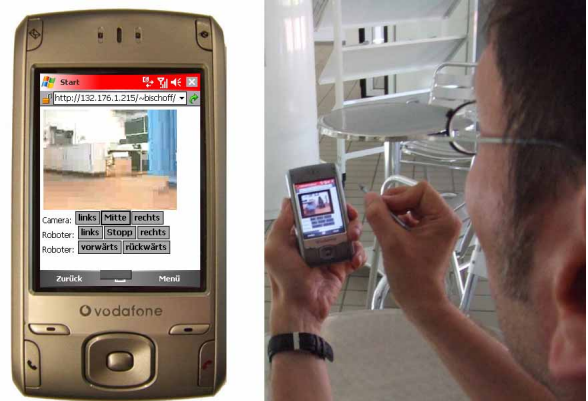


Figure 4: Pioneer 3AT mobile robot remotely controlled by Windows Mobile 2005 smartphone via GPRS

framework directly, but asynchronous HTTP requests are possible in Javascript. The new asynchronous web client operates on with each Windows Mobile device without any additional installed software (Fig. 4).

To support the mobile student in their experimentation practice a podcast variant of the required teaching material was derived automatically out of text based e-learning material by the use of text to speech techniques [6].

## VII. CELL PHONE SPEECH INTERFACE

Nevertheless a Windows Mobile device is required for this mobile experimentation practice. To establish an alternative interface for any cell phone a speech recognition based remote control was also realized.

The mobile robot is equipped with an Asterisk VoIP SIP telephony software [7] in this case. Since the SIP protocol is IP-based, no extra connection to a phone is necessary. A SIP service provider account is sufficient.

Asterisks features can be enhanced and configured by the so called 'dial plan extensions'. A custom dial plan has been developed to establish a voice menu with some speech recognition features by the help of the Asterisk speech recognition module ASR [8]. The dial plan extension was coupled via the Java server process to the robots low level control API to send movement commands directly to the underlying microcontroller. A benefit of this architecture is the reuse of the introduced (see IV) Java server application. A concurrent usage of the telephony service with other interfaces, like the web-based interface was enabled too. The speech recognition based voice menu robot user interface can be reached from every

phone worldwide by a land line phone number (fig 6). Such speech recognition interfaces to mobile robots will be very useful in robot human interaction in service robotics applications of the near future.



Figure 6: Pioneer 3AT mobile robot controlled by speech via cell phone

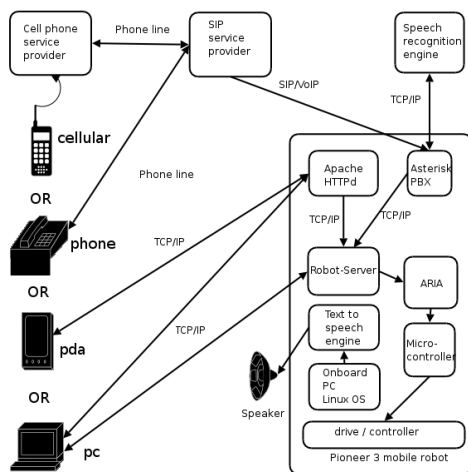


Figure 7: Speech interface communication structure

## VIII. FUTURE WORK

Future improvements will cover the integration of 3GP video streaming to establish a speech control with video feedback with every video telephony enabled UMTS 3GP cell phone. Support for other successful branches of smartphone operation systems like SymbianOS or the future Google Android cell phone OS will be added.

## REFERENCES

- [1] H. Hoyer, A. Jochheim, C. Röhrig, and A. Bischoff. Multiuser environment for a teleoperated laboratory. In Proceedings of the 1th IFAC Conference on Telematic Application in Automation and Robotics, Weingarten, Germany, July 2001.
- [2] N.N. SWIG, <http://www.swig.org/>, 2004.
- [3] A. Bischoff. M-learning with remotely operated laboratories. In 7th International Conference 'Virtual University' VU'06, pages 146-148, Bratislava, Slovak Republic, December 2006.
- [4] RFC 2616, <ftp://ftp.isi.edu/in-notes/rfc2616.txt>, 1999.
- [5] Jesse James Garret: Ajax: A New Approach to Web Applications, <http://www.adaptivepath.com/ideas/essays/archives/000385.php>, 2005
- [6] Andreas Bischoff. Podcast based m-learning with pediaphon - a web based text-to-speech interface for the free wikipedia encyclopedia. In 7th International Conference 'Virtual University' VU'06, pages 173-176, Bratislava, Slovak Republic, December 2006
- [7] N.N. Asterisk pbx. <http://www.asterisk.org>, 2007.
- [8] M Dirk Tostman, [http://www.voip-info.org/wiki/view/Asterisk+cmd+ASR&view\\_comment\\_id=14619](http://www.voip-info.org/wiki/view/Asterisk+cmd+ASR&view_comment_id=14619), 2007.
- [9] N.N., Google Android, <http://code.google.com/android/>, 2008
- [10] N.N. FFmpeg, <http://www.ffmpeg.org>, 2008

## AUTHOR

**A. Bischoff** is with the University of Hagen, Department of Electrical Engineering, Control Systems Engineering group (PRT), P.O. Box 940, D-58084 Hagen, Germany (e-mail: andreas.bischoff@fernuni-hagen.de).